

ВИЩА ОСВІТА

HIGHER EDUCATION

УДК 378.1:004

С.Г. Антощук, д-р техн. наук, проф.,
Д.В. Ушаков, магистр,
Одес. нац. политехн. ун-т

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПОДСИСТЕМЫ “ДЕКАНАТ”

С.Г. Антощук, Д.В. Ушаков. Онтологічна модель підсистеми “Деканат”. Описана онтологічна модель підсистеми “Деканат” задля забезпечення адаптованості і незалежності даних від конкретної інформаційної системи. Розглянуто основні концепти і відношення, а також принципи використання цієї онтології в конкретних інформаційних системах.

Ключові слова: онтологія, OWL, модель, адаптованість.

С.Г. Антощук, Д.В. Ушаков. Онтологическая модель подсистемы “Деканат”. Описана онтологическая модель подсистемы “Деканат” для обеспечения адаптируемости и независимости данных от конкретной информационной системы. Определены основные концепты и отношения, а также принципы использования этой онтологии в конкретных информационных системах.

Ключевые слова: онтология, OWL, модель, адаптируемость.

S.G. Antoshchuck, D.V. Ushakov. Ontological model of a “Dean’s office” subsystem. The development of an ontology subsystem “Dean’s office” to ensure adaptability and independence of data of a particular information system is described. The main concepts and relationships, as well as the principles of using the ontology in specific information systems, are considered.

Keywords: ontology, OWL, model, adaptability.

Автоматизированные информационные системы (АИС) являются обязательным компонентом деятельности образовательного учреждения, инструментом, обеспечивающим и реализующим происходящие в нем процессы, поддержку принятия решений, определяет эффективность его функционирования в целом. В рамках вуза, как системы, существует подразделение, занимающееся структурным учетом, — подсистема “Деканат”. Сложность решения данной задачи во многом связана с постоянными изменениями условий работы, большими объемами данных, зависимых от этих изменений, ограниченными сроками основных процессов вуза — учебного года, семестра, а также большим числом разрозненных информационных подсистем в АИС [1]. Эти факторы предъявляют жесткие требования к архитектуре и принципам построения АИС вуза как средства поддержки широкого спектра информационных процессов вуза. Критическим в таких условиях является обеспечение адаптируемости АИС к постоянным изменениям и требованиям.

Адаптируемость — это характеристика, определяющая способность системы к развитию в соответствии с потребностями пользователя и информационными процессами [2]. Способность АИС к адаптируемости определяется на этапе ее проектирования. Одним из перспективных направлений в моделировании различных процессов является использование онтологий как спецификаций некоторой предметной области [3].

Известны современные подходы к обеспечению адаптируемости АИС вуза [1]:

— использование клиент-серверной архитектуры с единой базой данных (БД); недостатки — невозможно объединить все данные такой сложной системы, как АИС вуза, в одной БД и невозможно отказаться от уже существующих и работающих систем;

— применение различных БД с интеграцией данных на уровне кода программы, что создает сложности при сопровождении таких АИС — необходимость изменения структуры одной из БД приводит к изменению куда большего числа подсистем АИС.

С учетом этих недостатков предложено формализовать описания предметной области и правила работы АИС с использованием онтологии — спецификации концептуализации, где в качестве концептуализации выступает описание множества объектов предметной области и связей между ними [4]. Это позволяет менять функциональность АИС при изменении различных к требованиям к ее функционированию путем добавления или редактирования правил онтологии, при этом не требуется изменение программного кода. Кроме того, возможны анализ, накопление и повторное использование знаний в предметной области, полученных из разных источников, другими словами — обеспечение независимости данных от конкретной АИС.

Для описания онтологий используют онтологические языки (или языки описания онтологий): обычно декларативные, в основном обобщение фреймовых; основанные на логике первого порядка либо на дескрипционной. Наиболее перспективный — язык сетевых онтологий (Web Ontology Language — OWL) [4].

Существует три разновидности OWL:

— Lite, используемый для классификации иерархий и простых ограничений на свойства элементов иерархии;

— DL — для обеспечения максимальной выразительности при сохранении полноты вычислений и разрешаемости логических выражений в онтологии;

— Full — для обеспечения максимальной выразительности и синтаксической свободы без гарантий вычисления логических выражений в онтологии.

Предлагается модель наиболее критичного процесса, связанного с преподаванием и обучением, реализуемого в подсистеме "Деканат", спроектированная для обеспечения адаптируемости АИС за счет применения онтологии на языке OWL DL, обладающим большей выразительностью, чем OWL Lite. Немаловажное требование — возможность расширения созданной онтологической модели до предметной области деятельности университета.

Основные правила выбранного языка:

— базовые понятия: класс объектов, свойства классов и объектов классов, объекты (экземпляры классов);

— все классы онтологии являются дочерними по отношению к классу Все (Thing) и родительскими к классу Ничего (Nothing);

— классы могут быть частично и полностью определенными, первые описывают необходимые свойства объекта для его включения в класс, вторые — необходимые и достаточные;

— объекты взаимосвязаны друг с другом посредством свойств;

— если некоторое свойство связывает объект А с объектом Б, то обратное свойство связывает объект Б с объектом А;

— свойства могут быть функциональными, обратными функциональными, транзитивными, симметричными, асимметричными, рефлексивными, иррефлексивными;

— на свойства могут быть наложены ограничения, существования, общности, мощности;

— ограничения существования (some) описывают класс объектов, обладающих хотя бы одним свойством по отношению к указанному классу;

— ограничения общности (only) описывают класс объектов, обладающих данным свойством только по отношению к указанному классу и никакому более.

Вычисление логических выражений в OWL основано на ограничении — т.н. предположении об открытости мира, которое говорит, что нельзя предполагать, что что-то не существует,

пока это не указано явно. Такое ограничение называется закрывающим (закрывающей аксиомой) и выражается в одновременном наложении ограничений существования и общности и, по сути, является шаблоном проектирования онтологий.

Для создания онтологической модели проанализированы основные информационные процессы в деканате института компьютерных систем, выделены концепты — понятия, представленные в последствии в результирующей онтологии классами.

Рассмотрим основные классы, используя Манчестерский синтаксис для их описания [4].

Class: Student

EquivalentTo: Person

and (hasDiscipline some Discipline)

and (hasDiscipline only Discipline)

“Студент” (Student) — класс, представляющий концептуализацию термина “студент” — любой человек, изучающий какие-либо дисциплины. Это полностью определенный класс (EquivalentTo), который определен через принадлежность к классам со свойством “изучать дисциплину” (hasDiscipline) и наложенными ограничениями: существование (some) и общность (only). Другими словами, студентом является любой человек, который изучает хотя бы одну из дисциплин и ничего кроме них, т.е. человек изучающий, например, операционную систему Linux студентом не является, но человек, изучающий предмет “Системное программное обеспечение”, — студент.

Class: Group

EquivalentTo: (hasMembers only Student)

and (hasMembers min 5 Thing).

“Группа” (Group) — класс, концептуализирующий термин “группа”. К нему принадлежат любые объекты, включающие (hasMembers) как минимум 5 студентов.

Class: GroupStudent

EquivalentTo: Student

and (isMemberOfGroup only Group)

and (isMemberOfGroup exactly 1 Group).

Студент группы — студент, являющийся членом (isMemberOfGroup) какой-либо группы. Как видно из определения, студент может является членом только одной группы, что указано с помощью ограничения мощности (exactly).

Class: Debtor

EquivalentTo: Student

and (hasDebt min 1 Thing).

Должник — студент, имеющий (hasDebt) хотя бы одну академическую задолженность (долг).

Class: Stipendiary

EquivalentTo: Student

and (hasStipend exactly 1 Studentship).

Стипендиат — студент, получающий (hasStipend) стипендию.

Class: Teacher

EquivalentTo: Person

and (isTeaching only Discipline)

and (isTeaching min 1 Thing).

Преподаватель — человек, преподающий (isTeaching) хотя бы одну дисциплину.

Class: Discipline

EquivalentTo: (isLearnedBy some (Group or Student))

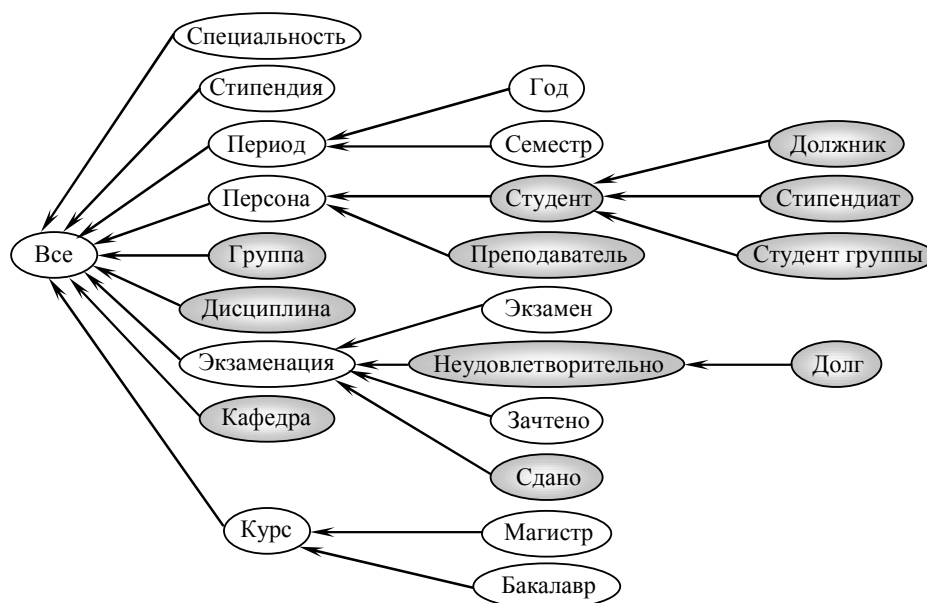
and (isLearnedBy only (Group or Student))

and (isTaughtBy some Teacher)

and (isTaughtBy only Teacher).

Дисциплина — класс объектов, изучаемых (isLearnedBy) группами или студентами, которые ведет (isTaughtBy) преподаватель.

Подобным образом были определены все классы онтологической модели подсистемы “Деканат”, после чего она была протестирована с помощью системы логического вывода Hermit [5]. С помощью редактора онтологий [6] визуализирована иерархия классов (см. рисунок, заштрихованы полностью определенные классы).



Иерархия классов онтологии подсистемы “Деканат”

Показаны базовые понятия, которые используются при работе деканата со студентами, в частности, для отслеживания успеваемости: сдача экзаменов, академические долги, стипендия. Онтология позволяет установить изучаемые дисциплины, кафедру и группу, к которой относится тот или иной студент, получаемую специальность.

Использование онтологии обеспечивает унификацию данных в различных АИС — для определения нового или изменения существующего понятия достаточно внести изменения в онтологию, а не менять все АИС, которые используют данное понятие, т.е. не требуется изменять программный код каждой подсистемы — достаточно внести изменения в онтологию. Изменение онтологии ведет к изменению функционирования АИС, построенных на ее основе, тем самым обеспечивается адаптируемость АИС. Так, при увеличении количества студентов в группе достаточно изменить мощность свойства hasMembers класса Group предложенной модели. Использование онтологии обеспечивает мультиплатформенность и независимость АИС от используемого языка программирования — написанные на различных языках и с использованием различных программных компонентов они могут использовать одну и ту же онтологию.

Приведенная модель отвечает требованию обеспечения адаптируемости за счет применения онтологического подхода в ее разработке. Дальнейшее определение концептов и расширение описания предметной области позволяет расширить онтологическую модель до уровня университета, также возможно включение данной онтологии в существующую модель АИС университета за счет использования стандартного языка OWL.

Тестирование с помощью системы Hermit показало, что онтология подсистемы “Деканат” является вычисляемой и непротиворечивой, а значит, верной и пригодной для построения ИС на ее основе.

Литература

1. Шахгельдян, К.И. Теоретические принципы и методы повышения эффективности автоматизации образовательных учреждений на основе онтологического подхода / К.И. Шахгельдян // Пром. АСУ и контроллеры. — 2010. — № 1. — С. 54 — 65.
2. Лядова, Л.Н. Метамоделирование и многоуровневые метаданные как основа технологии создания адаптируемых информационных систем / Л.Н. Лядова // Advanced Studies in Software and Knowledge Engineering. — 2008. — Vol. 2, № 4. — P. 125 — 132.
3. Андреева, Н.В. Выбор методов и средств онтологического анализа стандартов информационной безопасности [Электронный ресурс] / Н.В. Андреева. — СПб: СПбГУ ИТМО, 2009. — <http://www.twirpx.com/file/391160/>. — 30.08.2011.
4. OWL 2 Web Ontology Language Primer [Electronic resource] / P. Hitzler, M. Krotzsch, B. Parsia and others. — Cambridge : W3C, 2004. — <http://www.w3.org/TR/owl-primer>. — 30.08.2011.
5. Shearer, R. HermiT: A Highly-Efficient OWL Reasoner / R. Shearer, B. Motik, I. Horrocks // CEUR Workshop Proc. — 2008. — Vol. 432. — P. 1 — 10.
6. Horridge, M. A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools [Electronic resource] / M. Horridge. — 2011. — Manchester : The University Of Manchester, 2011. — <http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/>. — 16.11.2011.

References

1. Shakhgel'dyan, K.I. Teoreticheskie printsipy i metody povysheniya effektivnosti avtomatizatsii obrazovatel'nykh uchrezhdeniy na osnove ontologicheskogo podkhoda [Theoretical Principles and Methods for Increasing the Automation Efficiency of Educational Institutions on the Basis of Ontological Approach] / K.I. Shakhgel'dyan // Prom. ASU i kontrollery [Industrial ACS and Controllers] — 2010. — # 1. — PP. 54 — 65.
2. Lyadova, L.N. Metamodelirovanie i mnogourovnevye metadannye kak osnova tekhnologii sozdaniya adaptiruemykh informatsionnykh sistem [Metamodeling and Multilayer Metadata as the Basis of Adaptable Information Systems Technology] / L.N. Lyadova // Advanced Studies in Software and Knowledge Engineering. — 2008. — Vol. 2, # 4. — PP. 125 — 132.
3. Andreeva, N.V. Vybory metodov i sredstv ontologicheskogo analiza standartov informatsionnoy bezopasnosti [Elektronnyy resurs] [Selection of Means and Methods for Ontological Analysis of Information Security Standards [Electronic resource]] / N.V. Andreeva. — St.-Petersburg, 2009. — Available at: <http://www.twirpx.com/file/391160/>. — 30.08.2011.
4. OWL 2 Web Ontology Language Primer [Electronic resource] / P. Hitzler, M. Krotzsch, B. Parsia and others. — Cambridge : W3C, 2004. — <http://www.w3.org/TR/owl-primer>. — 30.08.2011.
5. Shearer, R. HermiT: A Highly-Efficient OWL Reasoner / R. Shearer, B. Motik, I. Horrocks // CEUR Workshop Proc. — 2008. — Vol. 432. — P. 1 — 10.
6. Horridge, M. A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools [Electronic resource] / M. Horridge. — 2011. — Manchester : The University Of Manchester, 2011. — <http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/>. — 16.11.2011.

Рецензент д-р техн. наук, проф. Одес. нац. политехн. ун-та Тонконогий В.М.

Поступила в редакцию 30 июня 2011 г.